Quiz Solutions and Grading Guidelines

Module 1

Module 1 Solutions

The table should be

1.

Algorithm	***Best-Case***	***Worst-Case***	***Stable?***	***In-Place?***
Insertion	**n**	**n^2**	**Yes**	**Yes**
Merge	**nlogn**	**nlogn**	**Yes**	**No**
Quick	**nlogn**	**n^2**	**No**	**Yes**

2.

False (Partition runs on the recursive calls to smaller lists as well)
False (Lomuto's is linear time in all cases)
False (The recursive calls could be on lists of uneven sizes)
True (By argument in lecture / slides)

3.

Should either discuss how insertion sort is faster on small lists OR almost sorted lists. Should discuss how switching in these two cases will speed up the algorithm.

4.

An inversion is a pair of items in the list that are out of relative sorted order. In the worst-case every pair is an inversion and there are n(n-1) / 2 inversions to fix (0 inversions = sorted list). One swap between adjacent elements can fix at most 1 inversion per swap. So...any algorithm will need to fix n(n-1) / 2 inversions, one per swap for a runtime of Omega(n^2)

Module 2

Module 2 Solutions

1.
Recurrence relation solutions are:

Nlogn

n^2

N

Does not apply

n^2 log(n)

2.

True (One for upper bound and one for lower bound to prove Big-Theta)

False (Case 2 does not have the polynomially different requirement because the functions aren't different at all)

True (We discussed this in class)

3.
Case 3 applies. K = log_4(3) = 0.79…, so n^0.79+0.01 is O(nlogn)

4.
**Because this is a homework problem, we are not providing the solution.

Module 3

Module 3 Solutions

1.
True (We discussed in class how removing any fraction of the list still guarantees logarithmic time to hit a base case.)

False (Strassen's was efficient for having fewer subproblems, not for more efficiently combining them)

False (Guarantees we select something in middle 40%...see slides)

False (One is to find the median of the medians, the other is to recurse on one half of list to solve the problem)

True (Might have bad pivot value each time)

False (You need the solutions to left and right subproblems to define delta and do the strip, so no.)

2.
The second and fourth bullets both apply, so accept either answer (or both)

3.
Third bullet is correct
Fourth bullet is also correct
**Which of these is "correct" depends on your interpretation of "ALL the points in y-order". If you mean all the points in that recursive subsection of the list, then the third bullet is correct (algorithm won't work but is still nlogn). If you assume ALL means the complete original list of points, then the fourth bullet is correct (algorithm won't work and time complexity is increased)

4.
Because they are within delta of the dividing line and are the only points that could be part of a solution that crosses the divide.

5.
So we can utilize the "only check the next seven" strategy, only checking points that are nearby in y-coordinate and stopping early.

6.

Because any point within delta is guaranteed to be within the next 7 in the list due to the “boxing” argument we did in class.